ALFAR Page No.
Date:

AES Algorithm [Advanced Encryption Standard]

invented by NIST in 2001

- bit level transposition and substitution.
- 128 and 192 bit keys.

* applications :- Scanlink → FD
→ HD
[Secured Access Program] uses AES 128 bit alg

* 3 versions of AES.

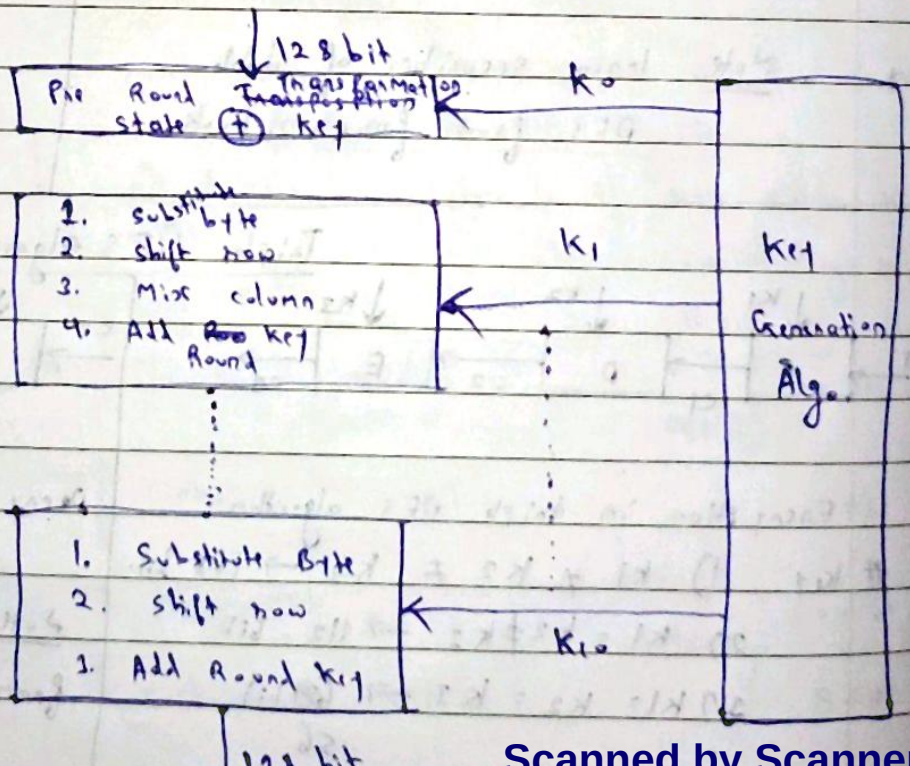
1) AES 128 → 128 bit key, 10 Rounds. ✓
2) AES 192 → 192 bit key No. of Round 12.
3) AES 256 → 256 bit 14 Rounds.

In syllabus only AES 128

- if forgot password, no recovery mechanism.
- In ios and ipad os uses AES. 128 bit.
- AES is lightweight, in Mobile phones or sensors.

Block diagram of AES Algorithm. 128 bit

State :- 4×4 matrix of plaintext and 4×4 key representation as matrix



Scanned by Scanner Go

Pre-round Transformation. state ⊕ Key

state. 4×4 matrix			
a_0	a_4	a_8	a_{12}
a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}

input:- ABCD... total 16			
A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

State = 128 bit

* In exam state Key and input will be 2×2 matrix

Q. Key :- DJSANGHVICOE Comp

D	N	I	C
J	G	C	O
S	H	O	M
A	V	E	P

4×4 Matrix Key

Note: it will be in uppercase character only, but if lowercase is given convert to uppercase.

state ⊕ Key

A ⊕ D

↓ ↓

00 03

0000 0000 ⊕ 0000 0011

0000 0011

= 0

13/03/24 * Key expansion in AES Algorithm *

128 bit Algo.

K_0 4×4. K_1

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

↓ ↓ ↓ ↓

w_1 w_2 w_3 w_4

K_0 (4×4)

c_0	c_4	c_8	c_{12}
c_1	c_5	c_9	c_{13}
c_2	c_6	c_{10}	c_{14}
c_3	c_7	c_{11}	c_{15}

↓ ↓ ↓ ↓

w_5 w_6 w_7 w_8

Scanned by Scanner Go

$R_{ij} = x^{i-1} \bmod \text{prime}$

$w_4 = w_0 \oplus g(w_3)$ ...

$w_5 = w_1 \oplus g(w_4)$.

$w_6 = w_2 \oplus g(w_5)$

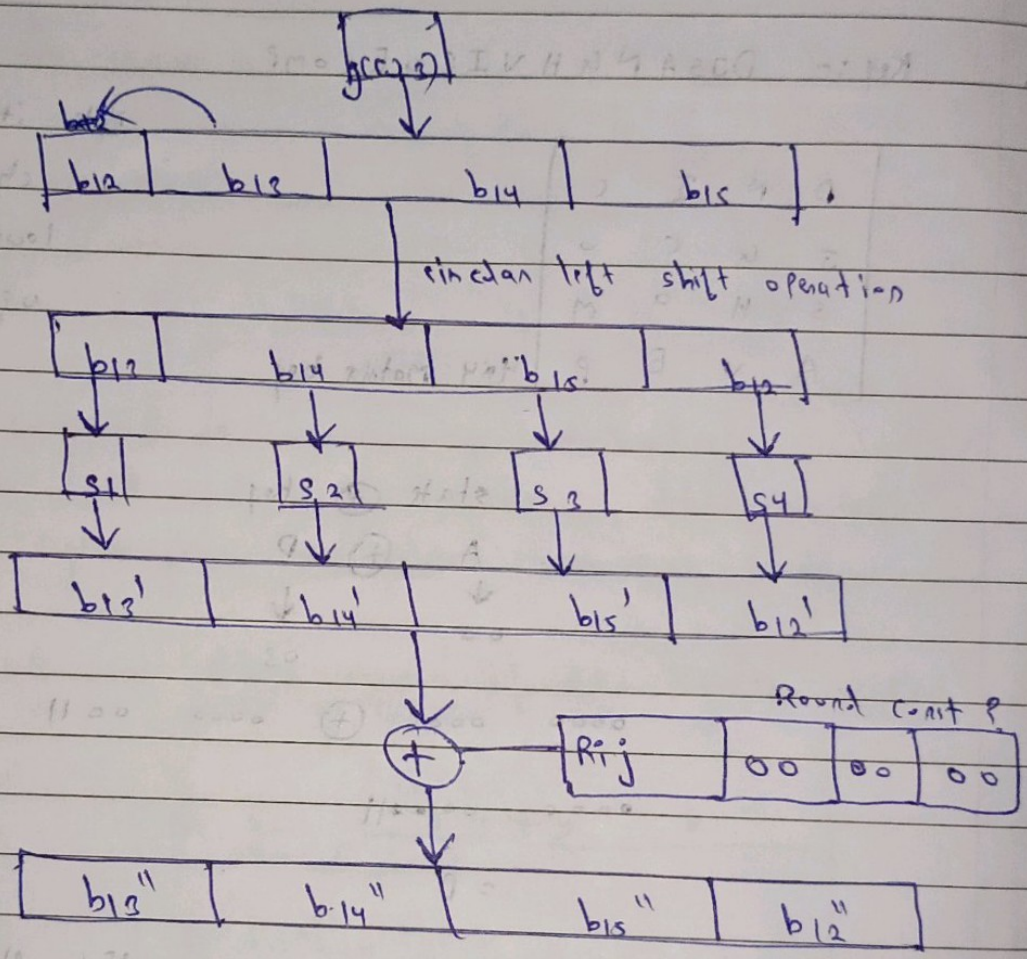
$w_7 = w_3 \oplus g(w_6)$

Round 1 = 01
Round 2 = 02
Round 3 = 04
Round 4 = 08
Round 5 = 10
Round 10^6 = 26

} Hexa

~~R_{ij} = Round~~

$w_3 = \{ b_{12}, b_{13}, b_{14}, b_{15} \}$



* using Key Expansion AES -128 bit technique generate w_4, w_5

given : $w_0 = \{ 24, 75, A2, B3 \}$
$w_3 = \{ 13, AA, 54, 87 \}$
$w_1 = \{ 34, 75, 51, 88 \}$

Scanned by Scanner Go

32 16 8 4 2 1

ALFAH Page No.
Date:

S-Box

13	AA	54	87
Ac	20	17	7D

→ Map this to this ← changed value

24 → 0010 0100
75 → 0111 0101
A2 → 1010 0010
B3 → 1011 0011
} w0

13 → 0001 0011
AA → 1010 1010
54 → 0101 0100
87 → 1000 0111
} w3

34 → 0011 0100
75 → 0111 0101
56 → 0101 0110
88 → 1000 1000
} w1

AC → 1010 1100
20 → 0010 0000
17 → 0001 0111
7D → 0111 1101

g(w3)

13	AA	54	87

→ Take w3

1 byte left shift circular

AA	54	87	13

→ 13 each on left

s-box

20	17	7D	AC

→ given

Round 1

Scanned by Scanner Go

x_1	x_2	x_3	x_4

$x_1 = 20 \oplus 01 = 21$

$x_2 = 17 \oplus 00 = 17$

$x_3 = 7D \oplus 00 = 7D$

$x_4 = AC \oplus 00 = AC.$

this we got $g(w_3) = 21, 17, 7D, AC.$

Now by formula.

$w_4 = w_0 \oplus g(w_3)$

↓

given.

$w_4 = \{24, 75, A2, B3\} + \{21, 17, 7D, AC\}$

$w_4 = \{05, 62, DF, 1F\}$ → 16 bit hexa decimal

$w_5 = w_1 \oplus w_4$ by formula.

$w_5 = \{34, 75, 56, 88\} \oplus \{05, 62, DF, 1F\}$

$w_5 = \{31, 17, 89, 97\}$

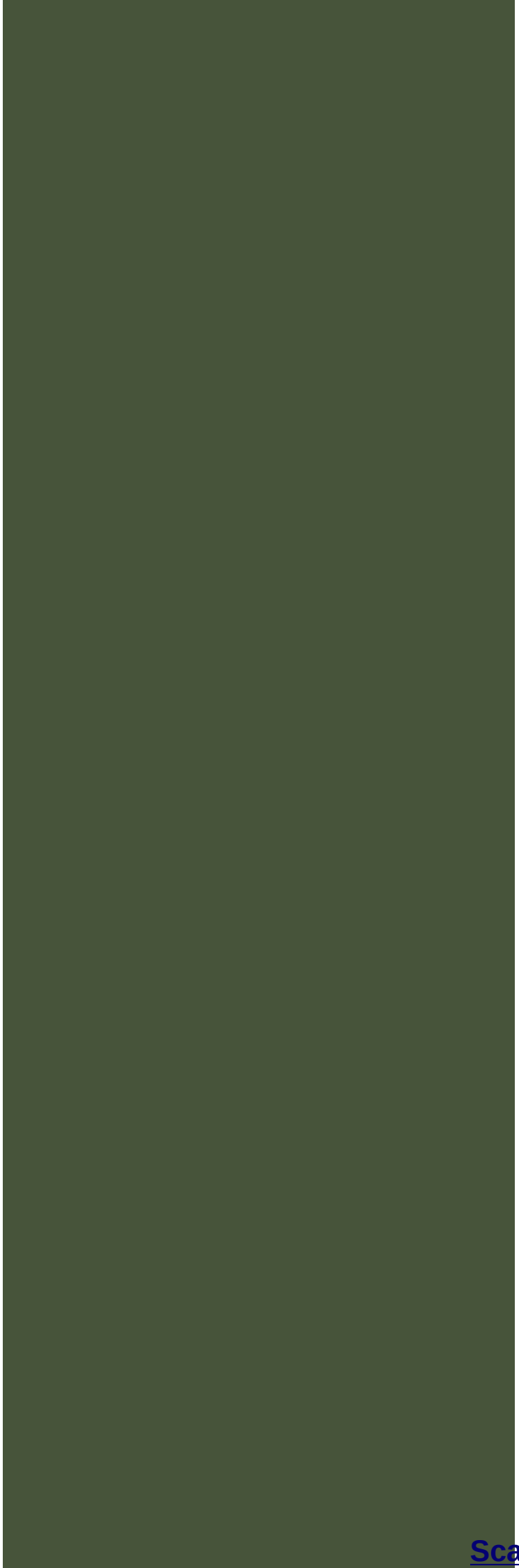
18/03/24

AES

- Sub byte.
- Shift Row
- Mix column
- Add Round key

sub byte. $\{2A, C1, 3B, 12\}$

Scanned by Scanner Go



Scanned by Scanner Go

4×4 Hexadecimal

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

remember only row 1, → then only CRS ←
1 byte circular RS.
1 byte circular RS.

if state = 4×4 then assume constant Matrix 4×4 if given or not given.

if if ~~then~~ state = 2×2 the constant matrix 2×2 if given or not given.

* Mix column operation numerical *

Given state = $\begin{bmatrix} 63 & 47 \\ f2 & 9c \end{bmatrix}$ constant matrix = $\begin{bmatrix} 02 & 03 \\ 01 & 02 \end{bmatrix}$

Resultant matrix b = To find :- ? $\begin{bmatrix} b_0 & b_2 \\ b_1 & b_3 \end{bmatrix}$ { output of column as transpose

Solution : Step 1 Constant matrix * column 1

$$\begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} 02 & 03 \\ 01 & 02 \end{bmatrix} * \begin{bmatrix} 63 \\ f2 \end{bmatrix}$$

$b_0 = (02 \times 63) \oplus (03 \times f2)$ ~~$\oplus (01 \times 63) \oplus (02 \times f2)$~~
row 1 × c1

$b_1 = (01 \times 63) \oplus (02 \times f2)$ row 2 × c1

No normal multiplication,
it is Finite Field Arithmetic operation.
for
GF (2^8) → Galiun Field
Maximum shifting of data / info.
will be computed by GF(2^8)

Scanned by Scanner Go

- In AES algorithm multiplication of two hexa decimal values is performed by using finite field. Arithmetic operation i.e known Field multiplicative operation. GF(2^8)

- This finite field Arithmetic operation is supporting maximum scrambling/ shuffling of data.

$b_0 = (02 \times 63) \oplus (03 \times f2)$

02 = 0000 0010

63 = 0110 0011

Next step finding polynomials values for the 8 binary data

GF(2^8) : default standard = {0000 0000 .... 1111 1111}

simplified GF(2^8) $x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$

{02} = 0 0 0 0 0 0 1 0

= x

{63} = 0 1 1 0 0 0 1 1

$x^6 + x^5 + x + 1$

$(02 \times 63) = x \times (x^6 + x^5 + x + 1)$

$= x^7 + x^6 + x^2 + x$ removing its binary

1 1 0 0 0 1 1 0 if present = 1

= C6 not present = 0.

$x^7 = 1$

$x^6 = 1$

$x^5 = 0$ not ... 0

$x^4 = 0$

$x^3 = 0$.

$x^2 = 1$

$(03 \times f2)$

03 = 0000 0011

f2 = 1111 0010

Scanned by Scanner Go

$x^7+x^6+x^5+x^4+x^3+x^2+x^1+1$

$0\ 0\ 0\ 0\ 0\ 0\ 1\ 1$

$1\ 1\ 1\ 1\ 0\ 0\ 1\ 0$

$x+1$

$x^7+x^6+x^5+x^4+x$

(02)x (f2)

$= x+1 \times (x^7+x^6+x^5+x^4+x)$

= (first multiply all by x then all with 1)

$= (x^8+x^7+x^6+x^5+x^2) + (x^7+x^6+x^5+x^4+x)$

(with x / with 1)

Now it is $x^8+x^7+x^6+x^5+x^2+x^7+x^6+x^5+x^4+x$

now cut the same terms

$x^8+x^7+x^6+x^5+x^2+x^7+x^6+x^5+x^4+x$

$G(x) = x^8+x^4+x^2+x$

The degree of above polynomial is 8 (x^8) which is not part of GF(2^8) and maximum degree supported by GF(2^8) is x^7 i.e. we need to convert above polynomial into reduce polynomial and for that purpose we have to divide above polynomial by irreducible polynomial.

irreducible polynomial is a fixed polynomial:

$x^8+x^4+x^3+x+1$

$P(x) = 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1$

$r(x) = G(x) = 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0$

Scanned by Scanner Go

ALFAH Page No.
Date:

10001 1011 | 1 0 0 0 1 0 1 1 0
- 1 0 0 0 1 1 0 1 1
0 0 0 0 0 1 1 0 1 1 → 0D
- 0 0 0 0 0 0 0 0 0 0

(03 × f2) = 0D

b0 = (02 × 63) ⊕ (03 × f2)
= C6 ⊕ 0D
b0 = CB

b1 = (01 × 63) ⊕ (02 × f2)
01 = 0000 0001
63 = 0110 0011

$GF(2^8) = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1$
{01} = 0 0 0 0 0 0 0 1
= 1

{63} = 0 1 1 0 0 0 1 1
$x^6 + x^5 + x + 1$

{01 × 63} = $x^6 + x^5 + x + 1$
= 0 1 1 0 0 0 1 1
b0 = 63

(02 × f2) 02 = 0000 0010
= f2 = 1111 0010
$GF(2^8)$
{02} = x
{f2} = $x^7 + x^6 + x^5 + x^4 + x$

Scanned by Scanner Go

$(02 * f2) = x \times (x^7 + x^6 + x^5 + x^4 + x)$
$= x^8 + x^7 + x^6 + x^5 + x^2$
=

$b(x) = x^8 + x^7 + x^6 + x^5 + x^2$
111100100

$P(x) = x^8 + x^4 + x^3 + x + 1$
$= 100011011$

$\frac{b(x)}{P(x)}$ $\frac{1111\ 00100}{1000\ 11011}$

```
              1
100011011 | 1111 00 100
          | -1000 11 011
          |  0111 11 111
          |
          |  b1 = (ff)
```

$(63 \oplus ff)$
=
0110 0011 XOR 1111 1111
= 9C

0110 0011
+ 1111 1111
1001 1100
9 C

~~b2 = (03 × 63) + (03 × f2)~~
with column 2

Constant matrix

$b2 = \begin{bmatrix} 02 & 03 \\ 01 & 02 \end{bmatrix} \times \begin{bmatrix} 47 \\ 9C \end{bmatrix}$

$b2 = (02 \times 47) + (03 \times 9C)$
$b3 = (01 \times 47) + (02 \times 9C)$

b2 = (02 × 47)

ALFAH | Page No.
Date:

02 = 0000 0010
47 = 0100 0111

GF(2^8) = $x^7+x^6+x^5+x^4+x^3+x^2+x+1$
0 0 0 0 0 0 1 0
{02} = x
{47} = x^6+x^2+x+1

(02 × 47) = $x(x^6+x^2+x+1)$

$= x^7+x^3+x^2+x$
$= 10001110$
b2 = 8E

b3 = (03 × 9C)
= 03 = 0000 0011
9C = 1001 1100

GF(2^8) = x x
{03} = $x+1$
{9C} = $x^7+x^4+x^3+x^2$

(03 × 9C) = $(x+1) \times (x^7+x^4+x^3+x^2)$

$= x^8+x^5+x^4+x^3+x^7+x^4+x^3+x^2$

$G(x) = x^8+x^5+x^7+x^2$
1 1 0 1 0 0 1 0 0
$P(x) = x^8+x^4+x^3+x+1$
1 0 0 0 1 1 0 1 1

$= \frac{G(x)}{P(x)} = \frac{1101001 00}{1000 1101 1}$

Scanned by Scanner Go

$$\begin{array}{r} 1 \\ 1101\ 00\ 100 \\ \oplus\ 1000\ 11\ 011 \\ \hline 0101\ 11\ 111 \\ B \quad F \end{array}$$

10001101 1

$b_2 = 8E \times BF$

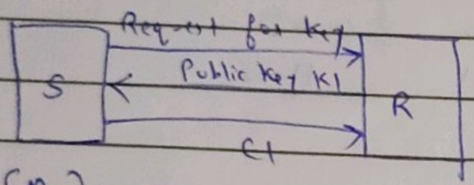
$b_2 = 31$

To find: b_3

02/04/24.

* Knapsack Cryptosystem. *

- Public key cryptography → Public key. / → Private key



$E(M)_{K1} = C1$

→ Decryption.

$D(C1)_{K2} = M$

RSA

K1 (Public) — K2 (private)

* RSA algorithm *

Step 1: Selecting 2 prime numbers P, Q

Step 2: $n = P \times q$

Step 3: Removing $z = \phi(n)$

$= (P-1) * (q-1)$ small

→ Eulan Totient Function.

Public key (e, n)

Private key (d, n)

Eulan Totient Function for non-prime number $\phi(10)$

ϕ 10

① writing Till that ϕ value

② removing the factors that comming into that